**[(转)Quirks模式与standards模式区别](http://www.cnblogs.com/jenry/archive/2010/11/21/1883068.html)**

建议：不推荐使用Quirks Mode。

Quirks Mode中发生了什么？  
Quirks Mode是一种浏览器（像IE，Firefox，Opera）操作模式。从根本上说，怪异模式（也称之为兼容模式）意味着一个相对新的浏览器故意模拟许多在旧浏览器中存在的bug，特别是在IE4和IE5中。

Quirks Mode是由文档类型探查法触发。也就是大家熟知的文档类型切换。这意味着浏览器检查一个HTML文档的开始，看它是否包含一个HTML规范所要求的文档类型声明。

Quirks Mode的目标是使旧页面显示出他们的作者想要的那样。旧页面可能利用旧浏览器已知的特性写成，或者至少是适应旧浏览器。更多关于怪异模式的信息请访问QuirksMode.Org。

Quirks Mode是什么没有权威的规范。毕竟，实质上Quirk Mode是一个故意违反CSS和HTML的规范。然而，因为作者们可能需要一个在Quirk Mode中实际上会发生什么的描述，所以我撰写了这个文档。

如果你有一个已经存在的页面，它看起来有效但是在开头缺乏一个文档类型声明（由HTML规范所要求）。但是你可能并不需要在开头简单地放置一个声明。原因是声明会使浏览器工作在一个和Quirks Mode相反的所谓的标准模式下。这可能意味着忽略其它所有东西。当你加了一个文档类型声明之后我会看到一个页面的所有内容消失了。更经常的，布局会以一种意料之外的方式改变。然而，如果你知道在Quirks Mode中会发生什么，改变的方式就不会在意料之外了。

在加一个文档类型声明之前，你应该使用验证器和检查器（validators and checkers）检查HTML和CSS代码的语法正确性。这可能还不够，因为页面仍然可能依靠一些在Quirks Mode下有效的东西写成。因此，你应该至少在IE7和Firefox2的标准模式下测试页面，例如，在加了一个文档类型声明之后，如果页面失效，下面列出的清单可能会对发现问题有用。

当创建一个新页面的时候，你不需要知道Quirks Mode，不需要经常想它。简单地根据HTML和CSS规范来写就行；这包括使用一个文档类型声明使较新的浏览器工作中标准模式下。再者，把文档类型声明放置在每个页面开头。因为如果有一些东西（甚至一条注释）在它前面，一些浏览器就会进入Quirks Mode。（如果你在使用XHTML，这会引起麻烦，但在更多案例当中，不管怎样你暂时不需要在web页面中使用XHTML。）但如果你决定使用一些可能在Quirk Mode下有效的特性，比如在table元素中使用一个height="100%"属性，你需要检查以下清单看是否有其它可能的含义。

在Quirks Mode下，浏览器已经遵守工作在下面的方式下，尽管并不是所有浏览器显示出以下所有特性：

l                box model是不正确的（与CSS规范有差异，尽管可能直观上更自然）。这意味着宽度和高度属性指定box元素包括padding和border的尺寸，不仅仅是元素的内容。（文档后面会给出一个demo。）

l                非置换内联元素的尺寸（比如，默认的span元素）受宽度和高度属性的影响。（在CSS规范中，它们会被忽略）。

l                元素的百分比高度（比如，HTML中的<table height="100%">或者CSS中的 table { height:100% }）是可行的，使用有效的高度作为引用，甚至当封闭快有自动的高度（默认的）。在标准模式下，高度取决于内容的要求；但是当封闭快设定了一个自己特定的高度值，百分比高度是有效的。

l                溢出被当做扩展box来对待。当一个元素的内容不适合指定的尺寸（明确的或隐藏的），那么溢出：可见的（默认）意味着当box尺寸作为指定，内容溢出。在Quirks Mode下，尺寸改变；这可以简单地看一个例子，如果box有一个背景颜色或一个border。

l                在Quirks Mode下，IE中根元素是body元素。根据规范，它是一个html元素。例如，在Quirks Mode下，为body设置一个margin是无效的。另一个例子，默认情况下，IE有一个垂直滚动条，尽管当没有东西可以滚动的时候，它是非活动状态（迟钝状态），在Quirks Mode下，你可以通过设置body { overflow: auto;}删除它（当不需要它的时候），但是在标准模式下，你仍然需要增加html { overflow: auto;}。

l                当在CSS中设置一个img元素或者input type="image"元素的时候，图像元素的padding属性会被忽略。

l                默认的浮动图片的水平margin是3像素（而不是0）。也就是说，如果img元素有align="left" 或者align="right"属性或者在CSS中有float: left 或者float: right应用与img元素，浏览器会认为就像img元素有hspace="3"属性（或者他的margin-left 和 margin-right属性值为3）。这是IE中的应用。在其它浏览器中，Quirks Mode可能会在图片的一边引起额外的margin，它的宽度是2像素而不是3像素。

l                图片的垂直对齐方式在某些特定情况下是在封闭快的底部，而不是文本的底线。当图片是元素中的唯一内容时就是这种情况，典型的是table的单元格。这意味着，比如，在Quirks Mode下，table单元格中的图片默认是在单元格底部（这经常是作者想要的），然而在标准模式下，在图片底部会有几像素的空间（除非有一种设置，比如，为img元素设置vertical-align: bottom）。

l                在CSS中，使用margin: 0 auto使一个块居中是无效的。注意：在IE（甚至IE7）中，为div在HTML中设置align="center"或者在CSS中设置text-align: center在块元素作为一个整体内部会不正确地居中，甚至在“标准模式”下。

l                字体属性不会从body或其他封闭元素继承到table中。特别是font-size。字体，颜色，行高也都有可能。比如，如果你设置body { font-family: Arial }，很有可能在table单元格中，字体属性仍然是浏览器默认。

l                在table单元格中字体大小的设置，一个百分比值被解释为相对于浏览器基本字体大小，而不是相对于封闭元素中（表单行）被CSS设定的字体大小。

l                字体大小关键字被不正确的解释，因此medium是比浏览器基本字体稍大，small和基本字体相等。类似的，整个比例关键字，xx-small, x-small, small, large, x-large, xx-large被逐个错误解释：每一个值被解释为比它本来大了一级。

l                畸形的属性值经常在猜测的基础上被解释。比如，margin: 10 作为 margin: 10px，color: ffffff 作为 color: #ffffff。在CSS中，这些违规的命令错误处理规则：语法上的错误应该被忽略。

l                在CSS中，字母大小写在类和标示符选择器中被认为是不重要的。因此，使用选择器.Foo匹配一个 or style="margin: 0cm 0cm 0pt 52.5pt; vertical-align: baseline; text-indent: -31.5pt; mso-list: l0 level1 lfo1; ">l                畸形的名字在类和标示符选择器中是被接受的。特别的，以一段或者一个数字符开始的名字（比如，在选择器中.123a and #42）是被接受的。

l                声明white-space：pre会被忽略。

l                固定位置不被支持：声明position: fixed被当做position: static（在IE7中）。

l                很多CSS中增加的支持（比如max-width属性和属性选择器）在IE7中，在Quirks Mode下不被使用。也就是说，有很多CSS中的特性在IE6中不被支持，在IE7中被支持，但仅仅是在标准模式下。查看微软博客请进入Details on our CSS changes for IE7。

l                标签杂烩解析。比如，如果一个文档包含标记<p>text<table>...</table>，Firefox中，在Quirks Mode下认为p元素包含table元素。在标准模式下，table的开始标签隐式地关闭了那个打开的p元素。不同之处在于，比如，你在p元素中设置了一个border。类似的，比如，Firefox接受在font中的ul。IE在这些情况中经常工作在错误的规则下，甚至在标准模式下，但是标准的一致性可以使用有效的标记来获得，经常使用明确的结束标签，像</p>，甚至当他们是正规的操作。

l                元素间的空白是重要的。比如，你有一个链接单。典型的你可以在表单项之间分行写li元素标记。（也就是说，在</li> 和 <li>标签之间）：

<ul> <li><a ...>...</a></li> <li><a ...>...</a></li> ... </ul>

然而，如果你设置display: block和为链接元素设置一个border，你可能会以项之间的垂直间隙作为结束。这会在IE7中Quirks Mode下发生，并且经常在之前版本的IE中发生。另一方面，在IE中Quirks Mode下，在以上标记中，如果你为li元素设置display: inline，那么在项之间就不会有空白。在标准模式下，在其它浏览器中，会有相对一个空白的空白。这被一些浏览器认为是标准一致性。

l                在Mozilla Firefox中，表单有一个默认的1em的bottom margin。（在IE中，两种模式下都有一个这样的margin。）这很显然是继承web浏览器的传统，在一个表单下面留下很多空间。怎样除掉一个表单下方的空白线过去是经常被问到的一个问题。

l                垂直margin在某些环境中是被禁止的，比如，当一个p元素作为首元素出现在一个td元素中。这或多或少是传统的行为，在IE7两种模式下，他被取代了。（在IE8中，只有在Quirks Mode下被取代）。

l                Wbr标记作为换行是很好的。浏览器很广泛的支持它。但是IE8中"标准模式"下缺不这样做了。这是坏消息，因为<wbr>标签一直是网页上其它非间断字符串中唯一有效的暗示换行方式。

l                Mozilla浏览器（像Firefox和Seamonkey）有一些附加的特性，在quirk.css归档。比如，table border的默认颜色是gray（在大多数浏览器中），在标准模式下使用与table前景色相反的颜色。在这些问题中，Quirks Mode确实是一个不同的兼容性模式。注意：在那个文档中，不是所有的设置都可以应用于所有版本的Mozilla。

所列的这个单子很可能是不全面的。这主要是与IE7相关。其它浏览器可能有模仿旧版本IE的与之程度差不多的Quirks Mode。

可以这样解释，在Quirks Mode下，width属性被（错误地）当做指定整个元素块的宽度，在这个例子中是8em。在标准模式下，它指定元素中内容的宽度，所以整个box的宽度为12em（加上边的宽度）。也就是说，总宽度包含左边和右边的空白，每个是2em。

在Firefox下，box模型在两种模式下可以正确应用。然而，你仍然可以在IE中Quirks Mode下，使用CSS命令或者在Web Developer Extention（它是一个在与CSS相关案例中很好用的工具）中使用Border Box模型模仿不正确的box模型。